



iDEAL SHOP INTEGRATION GUIDE

ASP .Net (C#) Merchant Plug-in

**composed by:
Friesland Bank N.V.**

version 1.0, January 06, 2009

© Copyright 2006, Friesland Bank N.V.



Table of Contents

Table of Contents	2
1 Introduction	3
2 Installation	4
2.1 Prerequisites	4
2.1.1 Installation	4
2.2 Global configuration	5
2.3 Security configuration	5
2.3.1 SSL- certificates	5
2.3.2 Signature - certificates	6
2.4 Demoshop Installation	8
3 Transactions	9
3.1 Directory request	9
3.2 Transaction request	10
3.3 Customer Redirect	10
3.4 Status request	11
3.5 Errors	11
4 APPENDIX A: Datacatalogue	12
5 APPENDIX B: Troubleshooting	13

1 Introduction

This document is intended to help software developers integrate the iDEAL payment method into their Microsoft ASP .Net (C#) webshop(s). This document describes the iDEAL Merchant Plug-in (the so called 'thinMPI') and offers installation instructions and some example shop code.

This document builds on the general introductory information. Readers not familiar with the general introductory information are advised first to absorb the information presented in the document 'iDEAL Shop Integration Guide – General Reference', so that detailed information about the iDEAL protocol and the integration process can be placed in the proper context.

2 Installation

This chapter describes all the steps necessary to install and configure the iDEAL Merchant Plug-in (thinMPI) and the example code.

2.1 Prerequisites

- Microsoft Internet Information Server
- Microsoft Web Server Enhancements version 1.0 SP1 (WSE 1), downloadable at <http://www.microsoft.com/downloads/details.aspx?FamilyId=06255A94-2635-4D29-A90C-28B282993A41&displaylang=en>
- Winhttpcertcfg.exe, part of the Windows Server 2003 Resource Kit Tools
<http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en>
- .Net Framework, downloadable at
<http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&displaylang=en>
- OpenSSL for creating your own certificate if not done yet (a Windows OpenSSL-Implementation can be found at <http://www.slproweb.com/products/Win32OpenSSL.html>, for all other platforms see <http://www.openssl.org>)
- Internet access through port 443 (SSL)
- Public key of acquirer (included in thinMPI)
- MerchantID and SubID
- thinMPI files copied to your development environment
- Access permissions (at least read permissions) for thinMPI files

2.1.1 Installation

- Install and configure the Internet Information Server according to the documentation from Microsoft.
- Download and install the Web Server Enhancements (Version 1.0 SP1) from the website given in the previous chapter.
- Download and install the Windows Server 2003 Resource Kit Tools.
- Download and install the Microsoft .Net Framework
- create a new “virtual directory” in your IIS
- put the thinMPI-Files in the virtual directory
- Open your Browser and start the DirReq.aspx

You should see a sample shop-page.

2.2 Global configuration

The following parameters have to be set in the file `Web.config` according to your environment. See APPENDIX A: Datacatalogue for a description of these parameters. Security related entries are explained in Chapter 2.3 Security configuration.

- `merchantID`
- `subID`
- `acquirerURL`
- `merchantReturnURL`

The merchant ID is the unique ID you got from your acquirer during the registration process. The subID identifies your contract and is usually 0.

The acquirerURL specifies the URL where all payment requests should be sent to. See the commentary in the file `Web.config` APPENDIX A: Datacatalogue for a list of all valid URLs.

The merchantReturnURL is the URL on your system that the customer is redirected to after the iDEAL payment.

2.3 Security configuration

This chapter describes all necessary security configurations relating the usage of certificates for SSL connections and to build signatures on messages.

2.3.1 SSL- certificates

The SSL certificate verifies that the system the merchant sends the request to is really the iDEAL system. By checking the SSL-Certificate the merchant makes sure that the remote system is indeed the system the merchant want to send the request to. Typically the acquirers SSL certificate is signed by an CA (e.g. VeriSign), which is already known and trusted by your system.

Only in case of the acquirers SSL certificate is not signed by an already trusted CA you have to add the SSL certificates manually to your system.

To do this the merchant's system have to have the public certificate of the iDEAL-System. Included in the thinMPI-Files there are two certificates. One for the test-environment and one for the production environment of iDEAL. To import these certificates perform the following steps:

- Open the Microsoft Management Console (mmc.exe)
- Choose File/AddRemove Snap-in...
- Press Add
- Choose Certificate from the list and press Add
- Choose Computer Account and press Next

- Win2k: Choose local Computer and press Finish
 - WinXP: Choose local Computer and press Finish
 - Win2k3: Use Network Services (can be changed through the user of application pool)
- Close all pop-up Windows
- open the tree Certificates and personal
- right click on all Tasks – Import
- choose the certificate type *.cer
- choose the certificate you want to import
- import the certificate to the folder Certificates – personal
- double-click on the imported certificate and click on certification path
- If the status is not “valid” import all certificates that your certificate belongs to in the folder “Trusted Root Certification Authorities”
- WinXP: Complete the following command whereby [myOrganization] can be found when opening the certificate in mmc (value of field “Issuer”).

```
winhttpcertcfg -g -c LOCAL_MACHINE\My -s [myOrganization] -a ASPNET
```

2.3.2 Signature - certificates

The signature is used to verify the authenticity of the request sent by the merchant. By verifying the signature, iDEAL checks

- whether the request really comes from the merchant identified by the merchantID
- whether the content of the request was somehow tempered

Although there is already a demo-certificate included in this package, it is strongly recommended to build own certificates at least for the production environment.

The certificate consists of two pieces: The private key and the public key. While the private key must be secured from reading by unauthorized persons, the public key will be uploaded to iDEAL.

In order to create a private/public key pair that can be used by ASP.Net the following steps have to be done:

- Download and install the openssl-suite.
- Generate a RSA private key:

```
openssl genrsa -des3 -out priv.pem -passout pass:myPassword 1024
```

- Create a new Certificate based on the private key:

```
openssl req -x509 -new -key priv.pem -passin pass:myPassword -days 3650 -out cert.cer
```

and enter the desired information.

- Create a PKCS12 Certificate that includes Certificate and Privatkey

```
openssl pkcs12 -export -in cert.cer -inkey priv.pem -out cert.p12
```

The next step is to import the newly created certificate into the registry. To accomplish that do the following:

- Open the Microsoft Management Console (mmc.exe)
- Choose File/AddRemove Snap-in...
- Press Add
- Choose Certificate from the list and press Add
- Choose Computer Account and press Next
 - Win2k: Choose local Computer and press Finish
 - WinXP: Choose local Computer and press Finish
 - Win2k3: Use Network Services (can be changed through the user of application pool)
- Close all pop-up Windows
- open the tree Certificates and personal
- right click on all Tasks – Import
- choose the certificate type *.p12
- choose the certificate (cert.p12) and mark the private key as exportable.
- import the certificate to the folder Certificates – personal
- double-click on the imported certificate and click on certification path
- If the status is not “valid” import all certificates that your certificate belongs to in the folder “Trusted Root Certification Authorities”
- WinXP: Complete the following command whereby [myOrganization] can be found when opening the certificate in mmc (value of field “Issuer”).

```
winhttpcertcfg -g -c LOCAL_MACHINE\My -s [myOrganization] -a ASPNET
```

There are two more steps to do:



- rename the value of the key Privatecert in your `Web.config` to [myOrganization] (can be found when opening the certificate in mmc: value of field “Issuer”).
- upload your certificate (*.cer) to the iDEAL System

2.4 Demoshop Installation

The example shop consists of three files that have to be copied into the directory `thinMPI\`:

- `DirReq.aspx`
- `TransReq.aspx`
- `StatReq.aspx`

Open the file `DirReq.aspx` in a browser. You now see a simple shopping system. You can enter quantities and prices for four different products.



Omschrijving	Aantal		Prijs		Totaal
Product 1	<input type="text" value="0"/>	x	<input type="text" value="0.01"/>	=	<input type="text" value="0.00"/>
Product 2	<input type="text" value="1"/>	x	<input type="text" value="7.15"/>	=	<input type="text" value="7.15"/>
Product 3	<input type="text" value="0"/>	x	<input type="text" value="8.95"/>	=	<input type="text" value="0.00"/>
Product 4	<input type="text" value="0"/>	x	<input type="text" value="10.00"/>	=	<input type="text" value="0.00"/>

Ordernummer:

 Totaalbedrag:

Figure 1 Example shop

3 Transactions

This chapter describes the three iDEAL requests:

- Directory Protocol
- Transaction Protocol
- Status Protocol

This chapter uses the example code to explain these three protocols. Each request is implemented in three separate example files: `DirReq.aspx`, `TransReq.aspx` and `StatReq.aspx` respectively.

3.1 Directory request

The file `DirReq.aspx` shows a (very simple) webshop in which the user can select some products.

By opening the file `DirReq.aspx` a Directory Request is automatically processed. This means that the most up-to-date list of iDEAL Issuers is requested from the acquirer. The selectbox below the orderform is populated with the results of the Directory Request.

Kies hier uw bank:

Figure 2 Selectbox populated after Directory Request

The following is done in `DirReq.aspx` to perform a Directory Request.

1. Create a `DirectoryRequest` object:

```
DirectoryRequest req = new DirectoryRequest();
```

2. Set parameters. Relevant parameters for the Directory Request are `merchantID`, `subID` and `authenticationType` (see APPENDIX A: Datacatalogue). These parameters are already set in the configuration file `Web.config`.

3. Create an instance of `thinMPI`:

```
ThinMPI mpi=new ThinMPI();
```

4. Process the request using the `thinMPI`:

```
DirectoryResponse res = mpi.processRequest(req);
```

Check the file `DirReq.aspx` to see how the result is handled to populate the selectbox.

Please note that the demoshop performs a Directory Request every time a user accesses `DirReq.aspx`. This is not mandatory, because the list of iDEAL issuers does not change very often. It is also allowed to perform the directory request periodically and store the list on your system. In 'iDEAL Shop Integration – General Reference' is described what errors may occur when the issuer list is not up-to-date.

After the user clicks the Order-button the form is posted to the file `TransReq.aspx`.

3.2 Transaction request

The file `TransReq.aspx` performs a Transaction Request based on the values it receives from `DirReq.aspx`. The process of performing a Transaction Request is very similar to performing a Directory Request:

1. Create Request object (`AcquirerTrxRequest` in this case)
2. Set parameters. Relevant parameters for the Transaction Request are set in the configuration file `Web.config`. The parameters `issuerID`, `purchaseID` and `amount*`, are obtained from the Directory Request.
3. Create an instance of `thinMPI`
4. Process the request using the `thinMPI`
5. Check and handle the result

* Please note that on the integration environment the parameter `amount` determines the result of the transaction. Use `amount=100` to simulate a successful transaction. See 'iDEAL Shop Integration – General Reference' for a description of the 7 (mandatory) test transactions.

3.3 Customer Redirect

After the Transaction Request the customer is redirected to the issuer. The URL of the issuer is provided in the response of the Transaction Request. It can be retrieved from the result of `ProcessRequest()` with the function `getIssuerAuthenticationURL()`:

```
//get IssuerUrl
string Url = res.IssuerAuthenticationURL;

//send redirect
Response.Redirect( Url );
```

3.4 Status request

After the transaction the customer is redirected back to the `merchantReturnURL`. To perform the mandatory (!) Status Request the steps below have to be completed. Read everything about the so-called 'collection duty' of the status of every iDEAL transaction in 'iDEAL Shop Integration – General Reference'.

1. Create Request object (`AcquirerStatusRequest` in this case)
2. Set parameters. Relevant parameters for the Status Request are set in the configuration file `Web.config`. The parameter `transactionIDssuerID` is obtained from the Transaction Request.
3. Create `thinMPI` instance
4. Process the request using the `thinMPI`
5. Check and handle the result

3.5 Errors

When an Error Response is returned instead of a normal response this is handled by the `thinMPI`. By calling the function `isOk` on the result of `ProcessRequest()` you can check if an `ErrorRes` is received. If so, the function `getErrorMessage()` returns a description of the error. For example:

```
if(!res.IsOk)
{
    // get Error Message
    lblError.Text = res.ErrorMessage;

    lblOK.Text = "No";
}
```

4 APPENDIX A: Datacatalogue

Parameter	Description
issuerID	ID of the issuer that the user has selected
merchantID	ID that identifies the merchant (you)
subID	Set to "0" unless you have specific reasons not to
authentication	Set to SHA1_RSA
merchantReturnURL	URL on the merchants system that the customer is redirected to after the payment. This page should perform the status request
purchaseID	Ordernumber from the merchant's system (your system)
amount *	Total amount for the order N.B. Please note that on the test environment the parameter amount determines the result of the transaction. See 'iDEAL Shop Integration – General Reference' for an overview of the 7 mandatory test transactions.
Currency	Set to EUR
ExpirationPeriod	Timeframe during which the transaction is allowed to take place. Notation PnYnMnDTnHnMnS, where every n indicates the number of years, months, days, hours, minutes and seconds respectively. E.g. PT1H indicates an expiration period of 1 hour. PT3M30S indicates a period of 3 and a half minutes. Maximum allowed is PT1H; minimum allowed is PT1M.
language	nl for dutch, en for english (used for showing errormessages is the preferred language)
description	Description of the order
EntranceCode	Mandatory code to identify the customer when he/she is redirected back to the merchantReturnURL
acquirerURL	The URL where the Directory, Transaction and Status Requests have to be send to. Integration environment: https://testidealkassa.frieslandbank.nl/ideal/iDeal Production environment: https://idealkassa.frieslandbank.nl/ideal/iDeal
privateKey	RSA private key generated by merchant (you)
privateKeyPass	Password used to generate Certificate based on privateKey
cert	Certificate generated with privateKeyPass and privateKey
certificate0	Default public certificate from acquirer (included with thinMPI)

5 APPENDIX B: Troubleshooting

Invalid signature or Authentication error

Make sure you uploaded the correct certificate via the iDEAL Dashboard and that the correct security parameters are defined in the configuration file `Web.config`

MerchantID unknown

Make sure you use either the demoshop `merchantID` (which was originally in the code) or your own `merchantID`. If the error occurs when using your own `merchantID`, your application is most probably not verified by the iDEAL Servicedesk.

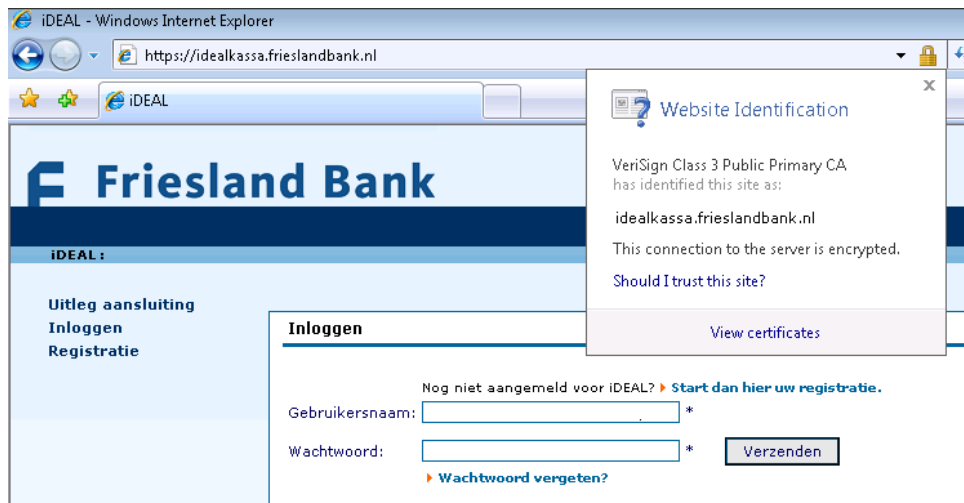
My Browser shows the aspx-Sourcecode instead of the html-page

If you did not have IIS installed when you ran the SDK or Visual Studio Setup, or if you uninstalled and reinstalled IIS after you ran the SDK or Visual Studio Setup, the file extension associations for `aspx` will not be in place. Visit the following website to gather more information and how to solve this: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q306005&GSSNB=1>

Acquirers SSL certificates expired

You can always download the current SSL certificates directly with your browser. Using Internet Explorer perform the following steps:

1. Open Logon page of your acquirer web site: <https://idealkassa.frieslandbank.nl>, click on lock visible near the URL (see screenshot) and choose “View certificates”.



2. In the certificate window goto tab “details” and choose “copy to file”. In the following dialog select base64 encoded .CER format and choose a filename. The current certificate will be stored on your computer.